

## Proyecto 2

Fecha de entrega: 9 de Junio, 2023

# Blockbuster-App: Procedimientos almacenados transacciones y *triggers*.

**Objetivo:** implementar la aplicación Blockbuster-App: CRUDs para el mantenimiento de Clientes, Películas, Préstamos y Categorías utilizando procedimientos almacenados y transacciones en JDBC.

### Requerimientos:

1. Agregar un nuevo proyecto Maven bajo su repository de GitHub: proyectos/blockbuster-app
2. Utilizar la base de datos “blockbuster” proporcionada como base de este proyecto.
3. Implementar el diseño de la base de datos “blockbuster” en MySQL.
4. Implementar CRUDs para los cuatro contextos de uso identificados en la base de datos:
  - a. Mantenimiento de clientes
  - b. Mantenimiento de películas
  - c. Mantenimiento de préstamos de películas
  - d. Lectura de bitácora de operaciones
5. **Para los mantenimientos de clientes y préstamos de películas:** los CRUDs deben de trabajar por medio de procedimientos almacenados para gestionar la selección, inserción, borrado y actualización de datos.
6. La creación y la forma de estos procedimientos almacenados quedan a criterio del estudiante: puede escribir procedimientos almacenados muy específicos o bien procedimientos almacenados que puedan llevar a cabo varias operaciones al mismo tiempo.
7. Todas las operaciones que involucren inserción, borrado y actualización, se deben ejecutar dentro de bloques transaccionales tanto en Java como en MySQL.
8. **Para el mantenimiento de películas:** utilice los métodos de JDBC para actualizar los datos de las tablas por medio de transacciones tomando como base esta guía: <https://docs.oracle.com/javase/tutorial/jdbc/basics/transactions.html>.
9. Para conectarse a la base de datos, debe de crear un usuario llamado “blockbusterappuser” y contraseña “blockbusterapppass” que solamente tenga acceso a la base de datos “blockbuster” y, dentro de esta, tenga privilegios de inserción, borrado, actualización y ejecución de rutinas (procedimientos almacenados).

- a. Este usuario es el que tiene que utilizar en su programa blockbuster-app para conectarse a la base de datos.
10. Debe presentar el esquema de toda su base de datos, registros de prueba y procedimientos almacenados en un archivo llamado "blockbuster.sql" que deberá guardar en un directorio llamado "db". Debe utilizar el programa "mysqldump" para esto.
11. El programa "mysqldump" no va a salvar la definición del usuario "blockbusteruserapp" por lo que, luego de haber obtenido el archivo .sql a partir de "mysqldump" agregue las líneas correspondientes a la definición del usuario "blockbusteruserapp" y sus privilegios.
12. De igual forma, una vez obtenido el archivo .sql por medio de "mysqldump" y de ejecutar el paso #10, agregue la siguiente línea: "SET autocommit = OFF;"

## Ejecución del programa

### Mantenimiento de clientes

```
> mvn exec:java -Dexec.args="clic [params]" // crea un cliente nuevo
> mvn exec:java -Dexec.args="clic Juan Soto juan@gmail.com 88669977" // Nombre,
apellido, email y número de teléfono
> mvn exec:java -Dexec.args="clir" // retorna todos los clientes
> mvn exec:java -Dexec.args="clir 11" // retorna cliente con ID 11
> mvn exec:java -Dexec.args="clid 11" // borra cliente con ID 11
> mvn exec:java -Dexec.args="cliu 11 Juan Soto juan@gmail.com 88669977" // actualiza el
registro en el ID 11. La llave primaria NO se puede actualizar.
```

### Mantenimiento de películas

```
> mvn exec:java -Dexec.args="movc [params]" // crea una nueva película
> mvn exec:java -Dexec.args="movc Matrix 1998-01-01 33 15" // nombre, fecha, cod.
categoría y unidades disponibles.
> mvn exec:java -Dexec.args="movr" // retorna todas las películas
> mvn exec:java -Dexec.args="movr 22" // retorna película con ID 22
> mvn exec:java -Dexec.args="movd 22" // borra película con ID 22
> mvn exec:java -Dexec.args="movu 22 Matrix 1998-01-01 33 15" // actualiza el registro en
el ID 22. La llave primaria NO se puede actualizar
```

### Mantenimiento de categorías de películas

```
> mvn exec:java -Dexec.args="catc [params]"
> mvn exec:java -Dexec.args="catc Fiction" // nombre de la categoría
> mvn exec:java -Dexec.args="catr" // retorna todas las categorías
> mvn exec:java -Dexec.args="catr 33" // retorna categoría con ID 33
> mvn exec:java -Dexec.args="catd 33" // borra categoría con ID 33
> mvn exec:java -Dexec.args="catu 33 Drama" // actualiza el registro en el ID 33. La llave
primaria NO se puede actualizar
```

### Mantenimiento de reviews

```
> mvn exec:java -Dexec.args="revc [params]" // crea un review
> mvn exec:java -Dexec.args="revc 33 11 4 'no me gusta' 2023-05-11" // código de
película, código de cliente, rating, texto de review, fecha del review
```

```
> mvn exec:java -Dexec.args="revr" // retorna todos los reviews
> mvn exec:java -Dexec.args="revr 55" // retorna el review con ID 55
> mvn exec:java -Dexec.args="revd 55" // borra el review con ID 55
> mvn exec:java -Dexec.args="revu 33 11 4 'no me gusta' 2023-05-11" // actualiza el registro en el ID 1. La llave primaria NO se puede actualizar
```

### Mantenimiento de préstamos de películas

```
> mvn exec:java -Dexec.args="loanc [params]"
> mvn exec:java -Dexec.args="loanc 33 11 <fecha de préstamo>" // Agregar la película 33 como préstamo al cliente 11, para el día <fecha de préstamo>
> mvn exec:java -Dexec.args="loanr" // retorna todos los préstamos
> mvn exec:java -Dexec.args="loanr 66" // retorna préstamos de la película con ID 66
> mvn exec:java -Dexec.args="loand 66" // borra el préstamo con ID 66
> mvn exec:java -Dexec.args="loanu 66 33 11 <fecha de préstamo>" // Actualizar el préstamo con ID 66 para la película 33 y el cliente 11, para el día <fecha de préstamo>
```

### Acceso a la bitácora

```
> mvn exec:java -Dexec.args="log 10" // retorna las últimas 10 (o N) operaciones registradas en la tabla "blockbuster_log" ordenadas por fecha (created_on) de forma descendente.
```

## Trigger para tabla de bitácoras

Cada vez que se ejecute una operación de escritura sobre las tablas de "rentals" y "reviews" se deberán ejecutar triggers de tipo "AFTER INSERT", "AFTER DELETE", "AFTER UPDATE" para insertar y actualizar la tabla "blockbuster\_log" con la actividad más reciente.

id	table_name	created_on	entry_text
1	rentals	2023-01-01	New rental for client id 22 and movie 33
2	rentals	2023-01-02	Update on rental 44
3	reviews	2023-01-02	Review 55 updated

## Reglas de negocio

En general las reglas del negocio deben de seguir y respetar el diseño de la base datos proporcionada para que se cumplan las siguientes condiciones:

1. No se puede agregar una película sin que exista una categoría antes.
2. No se puede borrar una película que esté prestada o que tenga reviews o préstamos asociados.
3. No se puede borrar un cliente que tenga reviews o préstamos (rental) asociados.
4. No se pueden borrar categorías que tengan películas asociadas.
5. No se pueden agregar préstamos sin que existan clientes o películas existentes.

# Evaluación

1. Proyecto Java/Maven que compila y corre (10pts)
2. Implementación de la base de datos (script con todo lo solicitado) (20pts)
3. Mantenimiento de clientes exitoso (20pts)
4. Mantenimiento de películas exitoso (películas, categorías, reviews) (20pts)
5. Mantenimiento de préstamos exitoso (20pts)
6. Lectura de la bitácora (10pts)

Total a evaluar: 100 pts.

## Entregables

El nuevo código bajo un tag llamado “proyecto-2”

